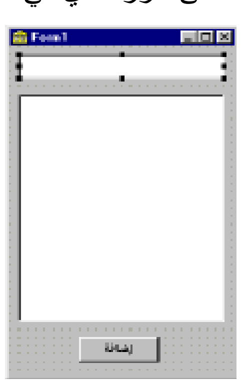
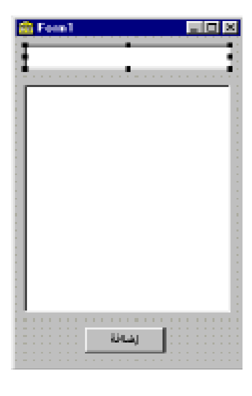
تعلم مع الأمثلة

المثال الأول:

لنضيف للمشروع خانة نص Edit ومفكرة Memo وزر ثم لننتقل لخاصة Caption للزر ونغيرها إلى "إضافة" وإلى خاصة Text لخانة النص ونمسحها ولخاصة Line للمفكرة ولنضغط على الزر الذي في خانة قيم هذه الخاصة ليظهر لك صندوق حوار، امسحه تماماً ثم اضغط Ok.





أضبط الآن خاصة Taborded لخانة النص على الصفر وللمفكرة على الواحد إن كانوا بخلاف ذلك،عندئذ سيغدو الشكل كالتالي:

انتقل الآن إلى حدث النقر للزر ثم ضع فيه السطر

```
Memo1.Lines.Add (edit1.text);
Edit1.Clear;
```

نفذ الآن التطبيق وكتب عبارة ما في خانة النص واضغط الزر.

لعلك أدركت الآن وظيفة المنهج Add التابع للخاصة Lines التابعة للكائن Memo.

هذا المنهج كما يظهر من وظيفته يقوم بإضافة سلاسل نصية إلى المفكرة، لعلك تلاحظ الآن أنّ التركيز يكون على خانة النص عند أول تنفيذ، ولكن في فيما بعد تغير التركيز إلى الزر عند الضغط عليه، وفي كل تجربة عليك أن تعيد التركيز بنفسك، ولتجاوز هذه العملية أضف السطر التالي للشفرة:

```
Edit1. SetFocus;
```

يعمل المنهج SetFocus على نقل التركيز للكائن المراد إعطاء التركيز له في التنفيذ.

الآن لو أردنا أن ننفذ هذه الشفرة بمجرد الضغط على المفتاح Enter دون استخدام الفأرة ما علينا أن نضبط خاصة Default للزر على القيمة True عندئذ يرتبط المفتاح Enter مع هذا الزر.

> يمكننا كذلك ربط المفتاح Esc مع زر ما من خلال الخاصة Cancel للزر وذلك بإعطائها القيمة True. وبالطبع فهاتين الخاصتين لا يمكن ربطهما مع أكثر من زر، أي يمكننا ربط الخاصة Enter مع زر ما وحيد، وكذلك الخاصة Cancel.

المثال الثاني:

لنضيف الآن إلى مشروع جديد زرين نغير خاصة Caption لهم إلى "تكبير" و"تصغير"، ولنضيف الكائن SpinEdit من الصفحة Samples، ثم لنضع في حدث النقر للزر الذي كتب عليه تصغير السطر التالي:

```
scaleby(spinedit1.Value,100);
```

وفي الأخر السطر التالي:

```
scaleby(100,spinedit1.Value);
```

نفذ الآن البرنامج وضغط هذين الزرين ولاحظ النتائج.

إن مهمة المنهج Scaleby هي التحكم بالكائنات الموزعة على الشكل حيث من خلاله يمكننا تكبيرها وتصغيرها وفق النسبة المئوية التي نريد، وذلك من خلال الوسيطين الممررين.

حيث أحد هذين الوسيطين في مثالنا هو عبارة عن القيمة التي توضع في الكائن SpinEdit وهي قيمة عددية حصراً يمكننا تغيرها كتابة أو بواسطة الأزرار الصغيرة التي رسم عليها أسهم والمرفقة مع الكائن.

> بالطبع عند زيادة القيمة داخل الكائن SpinEdit عن المائة فإن عمل الأزرار سينقلب، ويصبح التكبير تصغيراً والتصغير تكبيراً، ولمنع مثل هذه العملية علينا ضبط الحد الأعظمي والأصغري لقيمته، وذلك يتم من خلال الخاصيتين MaxValue و MinValue.

تعليق:

لعلك لاحظت أنه عند التنفيذ وتكبير حجم الكائنات، فإذا ما خرجت هذه الكائنات عند حدود الشكل فإنه سيظهر لهذا الشكل أشرطة تمرير، يمكننا التحكم بهذه الأشرطة من خلال الخاصة AutoScroll، فبإعطائها القيمة False نلغي ظهورها.

المثال الثالث:

لنضيف لمشروع جديد عنصر شبكة الألوان ColorGrid من صفحة Samples، والعنصر Panel من Stander، ولننقر نقراً مزدوجاً فوق خاصة اللون للشكل Color ليظهر لوح الألوان ثم لنختر منه اللون الأسود ليصبح لون الشكل أسود، وبعد ذلك لنضع في حدث النقر لشبكة الألوان السطرين التاليين:

```
Panel1.color: =colorgrid1. ForegroundColor;
Form1.color: =colorgrid1. BackgroundColor;
```

نفذ الآن البرنامج وضغط بأزرار الفأرة مرة بالزر الأيسر ومرة بالزر الأيمن ولاحظ النتائج.

عملياً إن الضغط بالزر الأيسر ينفذ المنهج الأول وهو يعطي اللون الأمامي للأشكال والذي أسندناه هنا للوح Panel، أما الضغط بالزر الأيمن ينفذ المنهج الثاني الذي يعطي اللون الخلفي وأسندناه للشكل.

لنتحدث الآن قليلاً عن الكائنين الجدد، ولنبدأ الحديث عن شبكة الألوان.

يمكننا تغير شكل عرض هذه الشبكة من خلال الخاصة GridOrdering، التي تملك مجموعة من القيم تتحكم بشكل خلايا هذه الشبكة دون تغير في العدد، وأما العنصر Panel فهو يعتبر من العناصر الحاوية، التي يمكن وضع عناصر أخرى غيرها بها كالأزرار وشابه ذلك، وتملك خاصتين Bevellnner و Bevelotuer اللتان تتحكمان بالإطار المحيط لهذا العنصر، كما يمكن التحكم بعرض هذا الإطار تم خلال الخاصة BevelWidth.

المثال الرابع:



لنضيف من الصفحة Stander عنصر المفكرة Memo و Panel وخمسة أزرار داخل Panel ولننتقل الآن إلى الصفحة Dialogs ولنختر منه العنصرين OpenDialog و SaveDialog.

لنمسح الآن خاصة Caption لعنصر Panel ولنغير نفس الخاصة للأزرار إلى الأسماء كما يتضح في الشكل.

ثم لنغير الخاصة Bevellnner للعنصر Panel إلى bvLowered والخاصة BevelWidth إلى الرقم 2.

لعلك الآن تفكر بالطريقة لتحديد الأزرار كاملة، من أجل ترتيبها من الخيار Alignment Palette في القائمة View، وتحاول أن تحيطها بإطار بواسطة الفأرة، لكنك ستلاحظ أنه بمجرد النقر فوق اللوح Panel ومحاولة إحاطة الأزرار بإطار فإن العنصر Panel سيتحرك مع الفأرة، والحل في رأيك الآن هو أن تختارهم عنصراً تلو الأخر بواسطة النقر فوقها بالفأرة مع الضغط على المفتاح Shift، ولكن الطريقة الأسهل هو أن تعود إلى الطريقة السابقة وهي الضغط والسحب ولكن هذه المرة مع الضغط على المفتاح Ctrl لتتم عملية التحديد داخل اللوح Panel بطريقة الضغط والسحب.

إن اللوح Panel هو واحد من مجموعة كائنات تملك قدرة احتواء عناصر غيرها، ومنها Groupbox و NoteBook و... والتي نستخدمها عادة عندما نريد تجميع عناصر لها وظائف متقاربة.

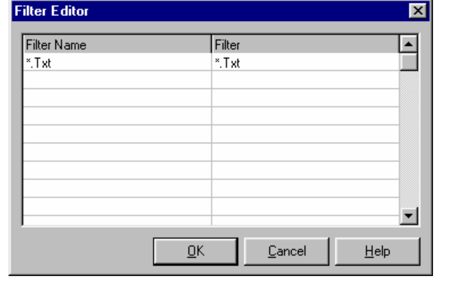
بقي عليك الآن أن تنظف عنصر المفكرة من الكتابة التي فيه كي تظهر كما في الشكل، وذلك من خلال الخاصة Lines، حيث نضغط على الزر الموجود في خانة قيم هذه الخاصة ليظهر صندوق حوار نمسحه ثم نضغط Ok.
بقية الآن أن نضيف الأسطر البرمجية لهذه الأزرار وهي كالتالي:
في حدث النقر لزر القص نضع memo1.CutToClipboard;
في حدث النقر لزر النسخ نضع memo1.CopyToClipboard;
في حدث النقر لزر اللصق نضع memo1.PasteFromClipboard;
في حدث النقر لزر الحذف نضع memo1.Clear;
في حدث النقر لزر تحديد الكل نضع memo1.SelectAll;
لاحظ أن تعامل عنصر المفكرة هو مع الحافظة Clipboard فينسخ لها ويأخذ منها، وأما بالنسبة لعملية الفتح والحفظ فتحتاج إلى روتينات أخرى سنذكرها الآ ن.
ضع الآن في حدث النقر لزر الفتح السطرين التاليين:

```
If OpenDialog1.Execute then;
Memo1. Lines.LoadFromFile(OpenDialog1.FileName);
```

في الحقيقة إن المنهج Execute هو عبارة عن دالة تعود بقيمة بوليانية، تكون هذه القيمة True إذا تم تهيئة صندوق الحوار بشكل طبيعي، وهو نفسه يعمل على تهيئة هذا الصندوق، ونحن الآن في هذا الشرط نختبر فيما لو كان قيمة هذا المنهج هي True أم لا ففي حال كانت True (أي التهيئة قد تمت بشكلها الصحيح) فإن صندوق حوار فتح سيظهر وتختر منه الملفات المراد فتحها.
بنفس الطريقة تتم عملية الحفظ وذلك من خلال السطرين التاليين:

```
If saveDialog1. Execute then
Memo1. Lines.SaveToFile(saveDialog1.FileName);
```

والذي يعمل على فتح صندوق حوار الحفظ.
نفذ الآن التطبيق وحاول أن تفتح الملفات النصية وحفظ ملفات أيضاً، سترى أن هناك نقص في هذا التطبيق وأن هناك أمور غير طبيعية تجري، فأنت تنشئ تطبيق لفتح الملفات النصية وحفظها أي الملفات ذات اللاحقة TXT، ولكنك تتفاجأ بأن صندوق حوار فتح يعرض أمامك جميع الملفات، كما أن عملية الحفظ يمكن أن تتم بأي لاحقة وبعد ذلك يصبح هذا الملف المحفوظ غير معرف، والحل...؟ لا بد أنه متوفر، وينبغي أن يكون سهلاً، ولكن هناك بعض الأمور التي تحتاج إلى مناقشة برمجية.
أنهي التطبيق إن كان في طور التنفيذ ثم اختر عنصر صندوق حوار فتح، وانتقل إلى الخاصة Filter التابعة له واضغط على الزر الذي يظهر في خانة قيم هذه الخاصة ليظهر لك صندوق له الشكل:



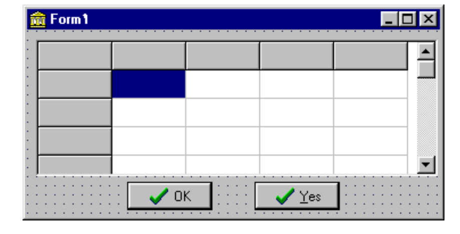
لعلك لاحظت من الشكل الصيغة التي يجب كتابتها في هذا الصندوق حتى تتم عملية التصفية، فيجب أن نخبره عن طريق Filter Name عن التوسع الذي يجب أن يظهر في خانة "الملفات" في صندوق حوار الفتح عند التنفيذ، ثم في خيار Filter من هذه النافذة يتم وضع نوع الملفات التي يجب فتحها في صندوق حوار الفتح وذلك بوضع اللاحقة المناسبة.
وبعد ذلك نضغط الزر Ok، جرب الآن تنفيذ التطبيق ثم

جرب الفتح و لاحظ الفرق.
إن تصفية عملية الحفظ تتم بنفس الطريقة، ولكنك ما تزال في عملية الحفظ تحتاج إلى كتابة اسم اللاحقة عند إجراء عملية حفظ، وسنناقش بعد قليل هذه العملية، ولكن قبل أن نبدأ بمناقشة هذه العملية، لنلاحظ أنه عند فتح ملف ما وكان هذا الملف كبيراً فنحن بحاجة إلى واسطة للتنقل في الملف، ولعل الواسطة الأكثر سهولة والتي تكاد تكون الوحيدة، هي أشرطة التمرير.
انتقل الآن إلى خصائص عنصر المفكرة واختر منها الخاصة ScrollBars، ولنلقي نظرة على قيم هذه الخاصة ستلاحظ أنه يملك أربعة قيم إما أن تلغي أشرطة التمرير، أو تختر شريط أفقي أو العمودي أو الاثنين معاً، وفي حال اختيارنا شريط عمودي فيجب إعطاء الخاصة WordWorp (أو ما تسمى بالالتفاف التلقائي) القيمة True والتي تعمل على ضبط النص بما يناسب عرض المفكرة.
الآن لنعود إلى مسألة اللاحقة التي نحتاج إلى كتابتها، إن مناقشة كتابة اللاحقة أم لا ليس بالمسألة الصعبة ولكن نحن نحتاج إلى بعض التركيز.
أنظر إلى الأسطر التالية التي يجب وضعها بدلاً من السطر في زر الحفظ السابق.

```
If saveDialog1. Execute then
Begin
    If Copy (Uppercase (saveDialog1.FileName),length(saveDialog1.FileName)-
3,4)='.TXT' then
        Memo1. Lines.SaveToFile(saveDialog1.FileName)
    Else
        Memo1. Lines.SaveToFile(saveDialog1.FileName+'.TXT’)
End؛
```

بعد أن تأكدنا من جاهزية صندوق حوار الحفظ بواسطة الشرط الأول، وجب علينا التأكد من وجود التوسع أو ما نسميه اللاحقة في الاسم، فإذا كان المستخدم قد كتب التوسع فيجب أن لا نضيفه وإلا فعلينا إضافته، وهذا ما قمنا به بالضبط في هذه الشفرة، فقمنا بأخذ الأحرف الأربعة الأخيرة بواسطة الدالتين Copy و Length وحولنا هذه الأحرف إلى الحجم الكبير، ثم قارنا هذه الأحرف إن كانت هي TXT. فعندئذ يكون المستخدم قد أدخل التوسع فلا حاجة لنا بإضافته، وإلى فعلينا إضافة التوسع.
وأما تحويل الأحرف إلى الحجم الكبير فذلك خشية أن نقارن مع الأحرف الصغيرة ويكون المستخدم يكتب بالحرف الكبير فعندئذ لن تتم المقارنة بشكل سليم وسيضاف التوسع حتى لو تمت كتابته من قبل المستخدم، فعند التحويل كل المدخل إلى حرف كبير والمقارنة معه نتخلص من هذه المشكلة.

المثال الخامس:
ابدأ مشروعاً فارغاً وأضف إليه من الصفحة Additional عنصر الشبكة StringGrid1 وزرين من نفس الصفحة الذي يحمل الاسم BitBtn، ولنضبط خاصية Kind للزرين، أحدهم على القيمة BkYes والثاني على القيمة Bkok، ولنلاحظ تغير شكل الزرين.
لنضع الآن في الزر الأول الشفرة التالية:




```
For i: =1 to StringGrid1.ColCount Do
    StringGrid1.cells [i, 1]: =inttostr(I);
For i: =1 to StringGrid1.RowCount Do
    StringGrid1.cells [1,i]: =inttostr(I);
```

وفي الزر الثاني الشفرة التالية:

```
For i: =1 to StringGrid1.RowCount Do
    For j: =1 to StringGrid1.RowCount Do
        StringGrid1.cells [i, j]: =inttostr (i)+ ',' +inttostr(j);
```

نفذ الآن التطبيق وضغط الزر الأول ثم الثاني ولاحظ ما يجري في الشبكة.
إن عنصر الشبكة أحد العناصر الهامة في بيئة Delphi وتملك مجموعة من الأنواع ولكن نحن نتعامل الآن مع StringGrid، وهي نوع من الأنواع التي تتعامل مع السلاسل، أي صممت خلاياها وكأنها خانة نص، تملك هذه الشبكة عدد كبيراً من الخصائص يهمنا منها الآن ما يلي:

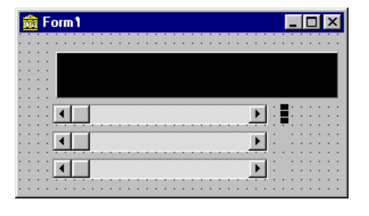
- ColCount: تعود هذه الخاصية بعدد الأعمدة في الشبكة.
- Fixedcolor: لضبط لون خلايا العناوين (أي خلايا السطر الأول والعمود الأول).
- FixedCols: لضبط عدد أعمدة العناوين.
- Col: (لا تتوفر في ضابط الكائنات وإنما أثناء تحرير الشفرة فقط) يعود برقم عمود ما.
- Cells: (لا تتوفر في ضابط الكائنات وإنما أثناء تحرير الشفرة فقط) يمرر لها وسيطان من نوع Integer يدلان على سطر وعمود الخلية المراد التعامل معها، ويسند لها قيمة من نوع String لتملئ الخلية المحددة.

> - نفس الخصائص تكرر للأسطر
> - جرب الخاصة الأخيرة بأن تضع في حدث النقر للشبكة السطر التالي:
>
> Form1.caption: =inttostr(StringGrid1.col);

إذاً فإن الشفرة السابقة الموجودة على الزر الأول تقوم بكتابة رقم ترتيب الأعمدة (الأسطر) في خلايا السطر الأول الموافقة (العمود الأول)، وأما الزر الثاني فهو يكتب في كل الخلايا ترتيبها عمودياً وأفقياً، وبما أنّ عملية الكتابة تتم بحلقة For والعداد من نوع Integer، وخلايا الشبكة لا تقبل إلى String لذلك نحن نحتاج إلى تحويل هذا العدد إلى سلسلة، وذلك يتم عن طريق منهج التحويل IntToStr الذي يمرر له وسيط عددي ليعود به على شكل سلسلة.
يمكننا إجراء عملية التحويل هذه بشكل معاكس بواسطة المنهج StrToInt الذي يحول السلسلة النصية إلى أعداد (بالطبع هذه السلسة يجب أن تكون تتالي أرقام وليست محارف أخرى).
يوجد عملياً عدداً لا بأس به من مناهج التحويل هذه والتي منها:
StrToFloat: وهو الذي يحول من سلسلة إلى نمط حقيقي وعكسه FloatToStr.
StrtoDate: وهو الذي يحول من سلسلة إلى نمط تاريخ وعكسه DateToStr.
StrtoTime: وهو الذي يحول من سلسلة إلى نمط وقت وعكسه TimeToStr.
وهناك صيغ أخرى عديدة من أجل إجراء التحويل القسري، ولكن ليس المقام هنا ذكرها جميعاً.
بقية علينا في هذا المثال الإشارة للكائن الجديد الذي تعاملنا معه، وهو هذا الزر الغريب من نوعه BitBtn، يملك هذا النمط من الأزرار كل مقومات الزر العادي الذي تعاملنا معه حتى الآن في تطبيقاتنا السابقة، بالإضافة إلى ثلة من الخصائص الإضافية التي قد نحتاجها في تعاملنا مع الأزرار، فهو مثلاً يملك مجموعة من الأزرار الجاهزة للتعامل معها في تطبيقاتنا والتي نختارها من الخاصة Kind، وكما نلاحظ في المثال منها زر Ok وزر Yes …
كما أنه يمكننا إضافة رسم ما لهذا الزر من خلال الخاصية Glyph والتي تملك في خانة قيمها زراً بالضغط عليه يفتح صندوق حوار فيه زر Lode منه يمكننا تحميل الصورة التي نريد ومن ثم نضغط الزر Ok لوضع الصورة المختارة على الزر.

المثال السادس:

لنبدأ مشروع فارغ ولنضيف إليه عنصر Panel وثلاثة أشرطة تمرير ScrollBar وثلاثة لافتات Lapel من الصفحة Stander، ثم لنغير لون Panel إلى الأسود ولنضبط خاصة Max لأشرطة التمرير على 255 ولنمسح خاصة Caption للافتات، حتى يبدو الشكل كالتالي:



الآن لنستخدم الحدث Change التابع لشريط التمرير الأول، ولنضع فيه السطرين التاليين:

```
Panel1.color:
=RGB(ScrollBar1.Position,ScrollBar2.Position,ScrollBar3.Position);
Label1.caption: =inttostr(ScrollBar1.Position);
```

لنكرر الآن هذه الأسطر في نفس الحدث لباقي الأشرطة مع تغير رقم اللافتة وشريط التمرير في السطر الثاني من الشفرة السابقة بما يناسب رقم الشريط الموجود فيه.

بعد الانتهاء نفذ التطبيق وحرك أشرطة التمرير ولاحظ تغير لون Panel، وكذلك تغير عنوان اللافتة.

لنبدأ الآن بشرح النقاط الأساسية في هذا المثال.

إن المنهج RGB يستخدم لمزج الألوان حيث يجب تمرير ثلاثة وسطاء عددية لهذا المنهج يمثل كل منها رقم تدرج لأحد الألوان الأساسية، حيث الوسيط الأول هو للون الأحمر والثاني للأخضر والثالث للأزرق، ونتيجة مزج هذا المنهج لهذه الألوان ينتج اللون النهائي، ولذلك مررنا الوسطاء لهذا المنهج على أنها قيمة موقع المنزلقة على شريط التمرير بحيث عند تغير موقع المنزلقة يتم تغير القيمة العددية وبالتالي يتغير لون Panel، ومن أجل أن يحدث التغير بشكل مباشر وضعنا النص البرمجي في الحدث Change لشريط التمرير حيث يقع هذا الحدث بمجرد إجراء تغير في موضع المنزلقة.

ولكن تدرجات الألوان التي تتعامل معها الحواسيب هي من 0 إلى 255 لكل لون من الألوان الأساسية، وبمزج هذه الألوان تظهر غالب ألوان الطيف، ومن أجل ذلك ضبطنا خاصة Max لأشرطة التمرير على القيمة 255 لكي لا تتجاوز المنزلقة هذه القيمة، والسطر التالي من الشفرة فليس له وظيفة إلاّ كتابة رقم التدرج في اللافتة.

لنتناول الآن الحديث عن بعض خصائص العناصر المستخدمة:

1- شريط التمرير: ScrollBar

- Kind: وتملك قيمتين إحداهما تجعل شريط التمرير أفقي والأخرى تجعله عمودي.
- LargeChange: يدخل بها رقم يضبط مقدار التغير الكبير لشريط التمرير، حيث التغير الكبير يقع عند الضغط داخل شريط التمرير وليس على الأزرار الخاصة لتحريك المنزلقة.
- Max: يوضع بها رقم لأكبر قيمة ستصل إليها المنزلقة.
- Min: يوضع بها رقم لأصغر قيمة ستصل إليها المنزلقة.
- Position: وهي لضبط موقع المنزلقة.
- SmallCange: يدخل بها رقم يضبط مقدار التغير الصغير لشريط التمرير، حيث التغير الصغير يقع عند الضغط على الأزرار الخاصة لتحريك المنزلقة.

2- اللافتة: Label

- Autosize: بضبطها على True تأخذ اللافتة حجم تلقائي يناسب النص الموجود بها.
- Transparent: بضبطها على True تصبح اللافتة شفافة.